Weekly Report

07/06/2015-07/12/2015

Research

TagClus

This week, I make a report about TagClus.In random walk model, we can compute the probability from node a to node b as follows:

$$\begin{cases} S_{k+1}(a,b) = \frac{\alpha}{|O(a)||O(b)|} \sum_{x \in O(a)} \sum_{y \in O(b)} S_k(x,y) & a \neq b \\ S_{k+1}(a,b) = 1 & a = b \end{cases} \tag{1}$$

with initialization S_0:

$$\begin{cases} S_0(a,b) = 0 & a \neq b \\ S_0(a,b) = 1 & a = b \end{cases} \tag{2}$$

where S_k means we can move from a to b in less than k steps(k steps from a and k steps from b). So S_k could be expressed as $\sum_{i=1}^{k} P_i(a,b)$. We define $P_i(a,b)$ as the probability reach from a to b in k steps.

Further, we need to calculate the similarity between two tags in TagClus. We believe that tags a is similar to tag b if they are used to tag the same resource. In our graph model(bipartite graph), the transition probability form a to b is nonzero when they can finally meet at resource in several steps. In bipartite graph, we can reach to resource from a tag in odd number steps. Fig.1 illustrate that tag 1 and tag 4 meet at resource 2 in 3 steps. We are supposed to add $P_i(a,b)$ when i is a odd number:

$$R_{2k-1}(a,b) = \sum_{i=1}^{k} P_{2i-1}(a,b)$$

Finally, the iterative formula of probability is

$$\begin{cases} P_{k+1}(a,b) = \alpha \sum_{x \in O(a)} \sum_{y \in O(b)} t_{ax} t_{by} P_k(x,y) & a \neq b \\ P_{k+1}(a,b) = 0 & a = b \end{cases} \tag{3}$$

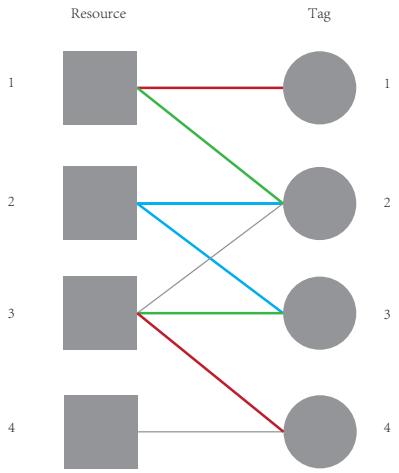

Figure 1: example caption

The difference between random walk and tagclus is that $P_{k+1}(a, b)$ is set to 0 in tagclus but $S_{k+1}(a, b)$ is 1 when tag a and b is same. In random walk we can walk from T1 to a and from T2 to b and $S_{k+2}(T1, T2)$ is nonzero because we can T1 and T2 can meet in 1 step which is forbidden in tagclus. It is the reason why we define S_k as the transition probability form a to b in less than k step and define $P_i(a, b)$ as the probability reach from a to b in k steps.

Mobility Pattern

I am trying to evaluate the result of k-means using silhouette coefficient(SC) this week. Given a object i who's silhouette coefficient is defined by Eq.4

$$SC(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{4}$$

$a(i)$ is average distance between object i and other object in the same cluster and $b(i)$ is average distance between object i and object in other clusters. The higher of SC, the better result of clustering. I add a new function in our java program to compute the average SC of every point.
However, SC is supposed to compute the distance between every two points leading a high computational complexity(n^2) which is much higher than k-means.I take a small sample to test because of the high complexity.As shownd in Table.1, It cost 20 minutes to compute SC when we input 10000 persons. We need hundreds of hours to calculate SC when there are 50 0000 persons which is intolerant. Besides, I try to find a best k of k-means for 1000

persons(24512 points), silhouette coefficient grows continually when k increases(Table.2). I think there are some bugs in my code.

Number of people	Time(s)
500	3.258
1000	12.919
2000	52.833
5000	328.79
10000	1303.754

Table 1: Time

Cluster number	SC
20	0.5922614053936373
80	0.7052977937398305
100	0.7232893454611151
120	0.7368427858345047
150	0.7492322561406840
200	0.7679698686585927
250	0.7804370695291505
300	0.7898459128826957
400	0.8046756611708712
500	0.8137676267726327
700	0.8293744191902724
1000	0.8437448464728854
1500	0.8588531116274699
5000	0.8884952831005910

Table 2: SC

Plan for next week

- Compute silhouette coefficient in matlab.
- Find other methods who's complexity is lower than (n^2) to estimate the number of cluster.